工程科学学报
Chinese Journal of Engineering

基于硬件虚拟化的云服务器设计与实现

郑臣明 姚宣霞 周芳 郑雪峰 杨晓君 戴荣

Design and implementation of a cloud server based on hardware virtualization

ZHENG Chen-ming, YAO Xuan-xia, ZHOU Fang, ZHENG Xue-feng, YANG Xiao-jun, DAI Rong

引用本文:
郑臣明, 姚宣霞, 周芳, 郑雪峰, 杨晓君, 戴荣. 基于硬件虚拟化的云服务器设计与实现[J]. 工程科学学报, 2022, 44(11): 1935-1945. doi: 10.13374/j.issn2095-9389.2022.01.12.005
ZHENG Chen-ming, YAO Xuan-xia, ZHOU Fang, ZHENG Xue-feng, YANG Xiao-jun, DAI Rong. Design and implementation of a cloud server based on hardware virtualization[J]. *Chinese Journal of Engineering*, 2022, 44(11): 1935-1945. doi: 10.13374/j.issn2095-9389.2022.01.12.005

在线阅读 View online: https://doi.org/10.13374/j.issn2095-9389.2022.01.12.005

工程科学学报, 第 44 卷, 第 11 期: 1935−1945, 2022 年 11 月
Chinese Journal of Engineering, Vol. 44, No. 11: 1935−1945, November 2022
https://doi.org/10.13374/j.issn2095-9389.2022.01.12.005; http://cje.ustb.edu.cn

基于硬件虚拟化的云服务器设计与实现

郑臣明[1)], 姚宣霞[1)✉], 周　芳[1)], 郑雪峰[1)], 杨晓君[2)], 戴　荣[3)]

1) 北京科技大学计算机与通信工程学院, 北京 100083　2) 海光信息技术股份有限公司, 北京 100193　3) 中科曙光信息产业成都有限公司, 成都 610041

✉通信作者, E-mail: yaoxuanxia@ustb.edu.cn

摘　要　随着互联网服务、大数据、云计算的兴起, 云服务器渐成需求主流. 相对于传统基于虚拟机的解决方案, 基于硬件虚拟化的云服务器因减少了软件的花销能更好地实现高效能、按需简约, 能更好地满足云计算的需求. 与传统云服务器相比, 该服务器的特点是高密度、高效能成本比、高效能功耗比和高可扩展性. 本文介绍了云服务器按需配置的设计理念、分布式硬件资源共享的系统结构和硬件资源虚拟化的方法. 设计并实现了一个基于硬件虚拟化的 16 个处理器的云服务器原型系统. 在该系统中, 基于现场可编程门阵列(Field programmable gate array, FPGA)设计实现云服务器的互联架构控制器(IFC). IFC 集成网络、存储和通用 I/O 资源, 为高密度的云服务器提供多处理器间的互联. 借助于 IFC, 所有 CPU 能够共享网络、存储和通用 I/O 资源, 实现硬件资源的虚拟化. 对原型系统的网络和存储性能进行了测试, 结果表明该系统不但具有传统云服务器的架构优点而且还提供更好的扩展性和更高的性能.

关键词　云计算; 云服务器; 共享存储; 共享网络; 共享 I/O

分类号　TP302.1

Design and implementation of a cloud server based on hardware virtualization

ZHENG Chen-ming[1)], *YAO Xuan-xia*[1)✉], *ZHOU Fang*[1)], *ZHENG Xue-feng*[1)], *YANG Xiao-jun*[2)], *DAI Rong*[3)]

1) School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

2) Haiguang Information Technology Co., Ltd., Beijing 100193, China

3) Dawning Information Industry Chengdu Co., Ltd, Chengdu 610041, China

✉ Corresponding author, E-mail: yaoxuanxia@ustb.edu.cn

ABSTRACT　Traditional cloud computing is developed from a high-performance cluster. Every server in the high-performance cluster has its own resources, including a CPU, memory, a network, I/O (Input/Output), a power system, and a heat dissipation system. Using software virtualization technologies such as the kernel-based virtual machine (KVM), Xen, VMware, and Hyper-V, these exclusive resources can be shared among these servers to improve the utilization rate. Although these technologies provide a great improvement in the resource utilization rate, some overhead in the process of software virtualization is inevitable. Server architecture and virtualization technology are the two factors that mainly affect cloud computing efficiency. With the rapid development of internet services, big data, and cloud computing, the cloud server has become mainstream instead of the traditional server. On the other hand, hardware virtualization technology has gradually developed. Compared with the traditional cloud computing solutions based on virtual machines, the cloud server based on hardware virtualization can achieve much higher efficiency to better meet cloud computing requirements by removing the software overhead. The cloud server's design concept of configuration on demand, distributed sharing of hardware resource architecture, and construction method of hardware resource virtualization are presented. A three-level interconnection

收稿日期: 2022−01−12

基金项目: 国家重点研发计划资助项目(2016YFB0200300); 国家重大科技专项“核心电子器件、高端通用芯片及基础软件产品”资助项目(2017ZX01028-102)

architecture of the cloud server is designed. In Level-1, the computing pool and the memory pool are built, while Level-2 is for the network pool, and Level-3 is for all resource pools. Different applications in these levels can be realized in the cloud server: Level-1 for computing-intensive applications, Level-2 for transactional applications, and Level-3 for virtual applications. A prototype system of a 16-processor cloud server using hardware virtualization architecture is designed and implemented. In this system, there are sixteen physical nodes. Every physical node is composed of a CPU and two DIMMs (dual inline memory modules). Different types of CPUs may be used in these physical nodes. Every four physical nodes form a computing module. In every computing module, a field-programmable gate array (FPGA)-based interconnection fabric controller (IFC) integrated network, storage, and general I/O resources is designed to interconnect these processors. All IFCs are interlinked. All the processors in this prototype system can share the network, storage, and general I/O resources to realize hardware resource virtualization through these IFCs. For the prototyping system, evaluation experiments on network performance tests by the Netperf program and storage performance tests by the FIO program are performed. The test results show that the prototyping system not only keeps the traditional cloud server's advantages but also provides better scalability and performance. The advantages of this cloud server are in providing a high-density, high performance-to-cost ratio, a high performance-to-Watt ratio, and high scalability compared with the existing traditional cloud server.

KEY WORDS cloud computing; cloud server; shared storage; shared network; shared I/O

云计算、大数据和人工智能被称为当前技术领域的三架马车，其中云计算[1]被看作继个人计算机变革、互联网变革之后第三次 IT 浪潮，已经深入到众多的技术领域中，逐渐影响人们生活的各个方面，出现了公有云、私有云、城市云、企业云、政务云、国资云、移动云计算等众多应用形式[2−3]. 云计算具有异构性、资源动态扩展、按需分配、资源池化等特点[4].

云计算正处于快速发展阶段，云计算的软硬件生态体系还在持续完善中，数据中心、云计算中心、互联网信息服务中心的云平台大多数呈现以高性能服务器为主体的构建特点，以高性能服务器[5]为物理节点，采用网络互联，通过在物理节点上构建虚拟机的方式提供云计算服务[6−9]. 这些物理节点一般都拥有自己的硬件资源，包括处理器、内存、存储、网络、I/O、供电、散热、管理等系统. 这种构建方式的特点是服务器的硬件资源利用率较低，空闲物理节点的硬件资源不能高效地释放给其他对硬件资源需求旺盛的物理节点，造成系统资源的浪费，如果集群系统上一定规模，那么这种资源浪费的现象就会更加严重. 统计报告显示，在没有采用虚拟化技术的系统中大部分服务器的 CPU 平均利用率只有 5% ~ 25%. 因此，如果能将服务器的全部硬件资源或部分硬件资源实现共享，通过负载均衡系统实现按需分配[10]，提高服务器的硬件资源利用率，那么就可以在不增加服务器数量或不增加服务器硬件资源的情况下，使系统整体服务能力得到显著提高，同时使系统整体功耗显著降低，从而显著提高系统性能功耗比.

云计算发展之初，重点是在满足应用需求，没有仔细考虑实际效能问题，但随着云计算逐步的发展成熟，不得不面对云计算的资源弹性、按需简约问题[11−12]. 为了解决这个问题有两种技术方向，一种是不断地设计或改进软件虚拟化技术[13−15]，另一种是设计或优化硬件资源虚拟化技术. KVM（Kernel-based virtual machine）、Xen、VMware、Hyper-V 是目前常用的软件虚拟化软件，基本原理是在物理裸机上运行相应的虚拟化软件从而模拟或虚拟出多个虚拟机，每个虚拟机在使用过程中就像真实地运行在物理机上一样. 利用多个虚拟机能够充分有效地利用和动态配置各种有限的硬件资源从而提高资源的利用率. 软件虚拟化技术发展得比较成熟，目前在进一步地提高云计算的资源利用率和性能上遇见了瓶颈，因为虚拟软件自身的运行毕竟会占用物理机的一部分资源. 在另一个技术方向硬件资源虚拟化技术方面，许多服务器公司已在自身产品的研发上或多或少实现了部分硬件资源按需简约[16]的功能. 这些服务器可概括为以下几类. SuperMicro 公司于 1997 年发布了 Twin 服务器，在 1 U 空间内放入了两台所有部件完全独立的双路服务器. 后续又出现了 2 U 4 节点、4 U 8 节点等多种形态，逐步采用了共享电源模块的方式来降低总体能耗，提升冗余性. 此类服务器提升了部署密度，但能耗下降有限. 在 2000 年出现了刀片服务器，在一个统一的机箱内，可以插入多个计算刀片，一个计算刀片就是一台服务器，同时所有服务器可以共享风扇、电源、网络交换、管理监控等模块，因此计算刀片上可以仅有处理器、内存和硬盘部件，设计上可以得到很大的简化. 刀片服务器具有管理比较方便、密度较

高、能耗较低的特点. 在 2009 年, Intel 公司提出了一个微服务器概念, 它是一种单插槽、可扩展、低功耗的入门级服务器, 具有高密度以及更高效的资源模块共享的特点[17]. 微服务器大多采用 Atom、ARM 等低功耗处理器[18], 当然也有时为了均衡计算性能, 会混合使用高性能处理器或者专用处理器[19]. 在 2011 年, 由阿里巴巴、百度、腾讯三方合作发起天蝎计划, 并在同年年底确立了最初的技术规范天蝎 1.0. 在 2019 年发展到天蝎 3.0, 天蝎计划旨在通过提出一种统一标准的设计规范, 实现低成本的可靠的灵活扩展, 目前对机柜子系统、节点子系统、供电子系统、散热子系统进行标准规范.

近些年来, 在云计算中出现了裸金属服务器(Bare metal server, BMS)的概念[20], 这是为了解决一些对计算和 I/O 性能要求高、数据处理量大的业务而提出的一种服务器, 利用它可使云计算能够应用于核心数据库、关键业务应用系统、高性能计算等场景. 裸金属服务器是硬件和软件优势相结合的产物, 本质上仍是一台传统的物理服务器, 但同时又具备云计算技术的虚拟化服务功能. 它是在只靠传统的软件虚拟化技术无法满足一些业务需求的情况下, 采用硬件资源独占的方式分配给这些业务使用来解决性能瓶颈[21]问题, 其实这种做法与云计算所追求的资源共享、按需分配的特点相悖. 为此, 阿里云对裸金属服务器进行了改进优化, 推出了弹性裸金属服务器, 即神龙云服务器[22]. 神龙云服务器兼具了物理机和云服务器的优点, 本质上仍是一台裸金属物理机, 即具有物理机的高性能、安全物理隔离的特点, 另一方面采用独立于物理机 CPU 的额外专用芯片[23](如 FPGA、Intel Xeon CPU)分担原物理机 CPU 的一些虚拟化任务, 如虚拟机监视器(hypervisor)所承担的管理调度、设备软件模拟等方面的任务. 神龙服务器同时利用额外的专用芯片桥接存储、网络等 I/O 设备, 协助物理机 CPU 实现 I/O 设备硬件虚拟化. 专用芯片能够运行阿里云的特有管理软件, 可有效地整合进阿里云已有的管理系统中, 实现快速的交付能力和实时的业务响应能力. 神龙云服务器架构有三个模块组成: 计算子板, I/O-Bond 桥接板, BM-hypervisor 底座. BM-hypervisor 底座作为基础物理服务器, 采用 Intel E5 系列 CPU 扩展出 16 个 PCIe(Peripheral component interconnect express)槽, 每个槽对应一个计算子板, 每个计算子板带有一个用 FPGA 设计的 I/O-Bond 桥接板用来连接计算子板和 BM-Hypervisor 底座. 每个计算子板就是一个独立的物理机, 上面有 CPU、内存、PCIe 总线和 I/O-Bond 桥接板. I/O-Bond 桥接板是用 FPGA 实现的硬件接口, 模拟实现 I/O 设备, 实现硬件虚拟化. BM-hypervisor 底座上的 CPU 负责 16 个计算子板的 hypervisor 功能, 并运行阿里云的管理软件. 神龙云服务器的架构特点决定了比传统的云服务器具有更高的计算能力、更高的虚拟化效率, 但其优势只能在阿里云体系下才能得到发挥, 离开了阿里云体系其架构缺点也十分的明显. 为了提高神龙云服务器的计算能力、虚拟化效率, 采用 16 个 FPGA 芯片和一个 Intel E5 CPU 作为专用的加速芯片, 用大量专用芯片换性能所付出的成本比较高, 对于小规模的云计算来说性价比不高, 只有在像阿里云这样成千上万规模节点的云计算中才能体现性价比的优势. 神龙服务器受限于 BM-hypervisor 底座上 CPU 的 PCIe 扩展能力不能容纳较多的计算子板, 单台神龙服务器能容纳 16 个低端 Xeon E3-1240 v6 计算子板, 或者 8 个中端 Xeon E5-2682 v4 计算子板, 或者 1 个高端 Xeon E5-8163 计算子板, 16 个计算子板几乎已是上限, 弹性的拓展能力受限. 计算子板的异构性是神龙服务器的一大特点, 但在单台神龙服务器内部同时只能使用一种规格的计算子板, 如更换需要同时更换.

针对如上云服务器的不足之处, 本文着眼于云计算系统内计算、内存、存储、网络以及其他 I/O 资源研究一种基于硬件资源的虚拟化、共享池化技术, 设计一种云服务器系统架构, 分别形成计算池、内存池、存储池、网络池以及 I/O 池, 每种池中的资源既可同构、又可异构. 根据本文提出的云服务器系统架构, 设计云服务器原型机, 并测试和评估网络池化性能、存储池化性能. 此系统能够根据实际负载要求动态从资源池中申领资源、释放资源, 使系统资源达到最大化的利用.

1 云服务器系统的架构设计

1.1 云服务器硬件资源池化方法

在传统的云平台架构中, 每个物理节点都是一个独立的服务器, 如图 1 所示, 物理节点拥有自己的硬件资源, 包括处理器、内存、存储、网络、I/O、供电、散热等系统, 这些硬件资源无法被其他物理节点所共享使用, 致使虚拟机 (Virtual machine, VM) 只能建立在自己的物理节点上, 而不能跨物理节点资源建立, 在负载不均匀的情况下会造成不同的服务器忙闲不均, 造成资源的极大浪费.

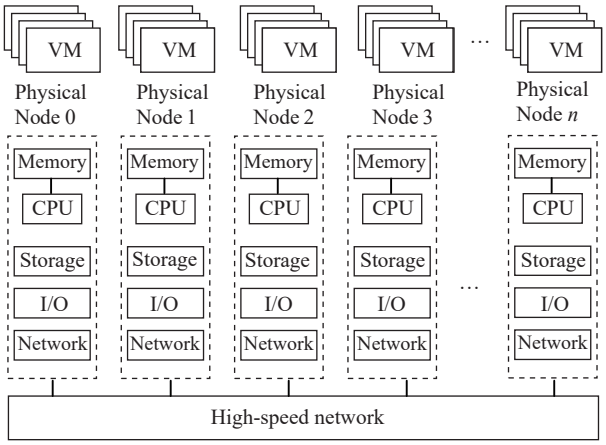

图 1　传统的云平台架构

Fig.1　Traditional cloud computing platform

云服务器所追求的一大特点是要构建计算、内存、存储、网络和 I/O 的共享架构. 服务器硬件资源共享的前提是将系统的计算、内存、存储、网络以及其他 I/O 资源聚合在一起, 分别形成计算池、内存池、存储池、网络池以及 I/O 池, 如图 2 所示. 每种池中的资源既可同构、又可异构.

为了实现服务器资源池化的目的, 服务器硬件资源接入一个高性能互联网络, 如图 3 所示. 一方面, 通过管理系统按照应用负载的需求, 有效地组织该物理节点的计算、内存、存储、网络和 I/O 按需配置, 另一方面, 要确保按需构建的物理节点的高效能. 客观上讲, 资源共享会存在一定的共享开销, 需要建立一种平衡的体系结构将各类资源的共享开销降到最低.

高效能的云服务器可以采用多级的高性能互联网络构建云服务器系统, 硬件资源根据不同的需求(如与处理器的耦合程度、性能要求)分布在互联网络的不同层, 这些硬件资源可以有本地宿主, 也可跨层协调给远程宿主.

如图 4 所示, 本文设计的云服务器高性能互联网络由三级构成, 以一级互联为单元构成二级互联系统, 再以二级互联为单元构成三级互联系统, 即全系统.

一级互联: 对应一个物理节点, 物理节点上可以有多颗 CPU 和多组内存, 主要实现 CPU 资源和内存资源的池化. 由于 CPU 和内存对物理节点性能有很大影响, 内存仅作为处理器的私有资源, 两者之间的连接必须是紧耦合关系, 采用专门的总线互联, 例如第四代双倍数据速率 (Double data rate 4, DDR4) 内存总线. 同一物理节点采用对称多处理 (Symmetric multi-processor, SMP) 等方式组成多路 CPU 的内存共享, CPU 之间通过超路径互联 (Ultra path interconnect, UPI) 总线、超级传输 (Hyper transport, HT) 等类型的高速互联总线连接, 采用高速缓存相关的非一致性内存访问 (Cache coherent non-uniform memory access, CCNUMA)、非一致性内存访问 (Non-uniform memory access, NUMA) 等架构/协议来实现内存资源的池化. 此互联域内资源非常适用于对计算性能敏感性的业务场景.

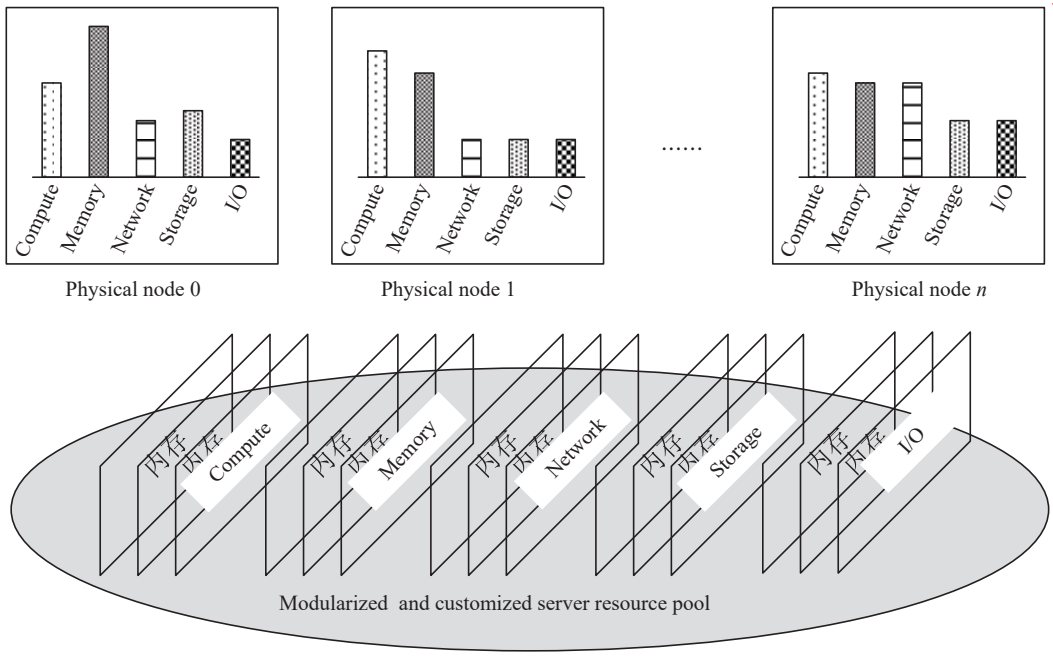

图 2　云服务器硬件资源池化示意图

Fig.2　Diagram of server pooling hardware resources

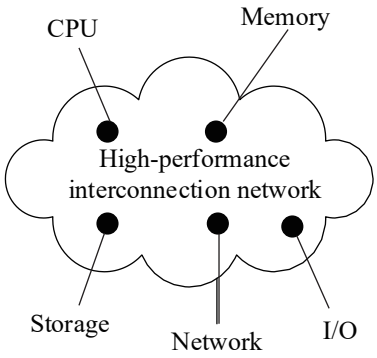

图 3 基于高性能互联网络的硬件资源池化方法

Fig.3 Hardware resource pooling approach based on a high-performance interconnection network

二级互联：以一级互联域对应的物理节点为单元建立二级互联网络，这是针对一定数量的处理器彼此共享同一物理资源的互联架构．该级互联虽对互联的扩展性无特殊要求但对性能要求却很高，以降低因共享带来的开销．该级互联一般采用 Crossbar 拓扑，在性能和吞吐量上满足共享要求．此级互联主要是针对存储、网络、I/O 资源的池化，为一级互联上的不同物理节点提供共享存储、网络、I/O 资源．二级互联避免了为每个物理节点单独设计私有资源而带来成本浪费的问题．与一级互联域配合使用，可解决需要大量处理器且对计算性能敏感的业务场景，如事务型数据库、

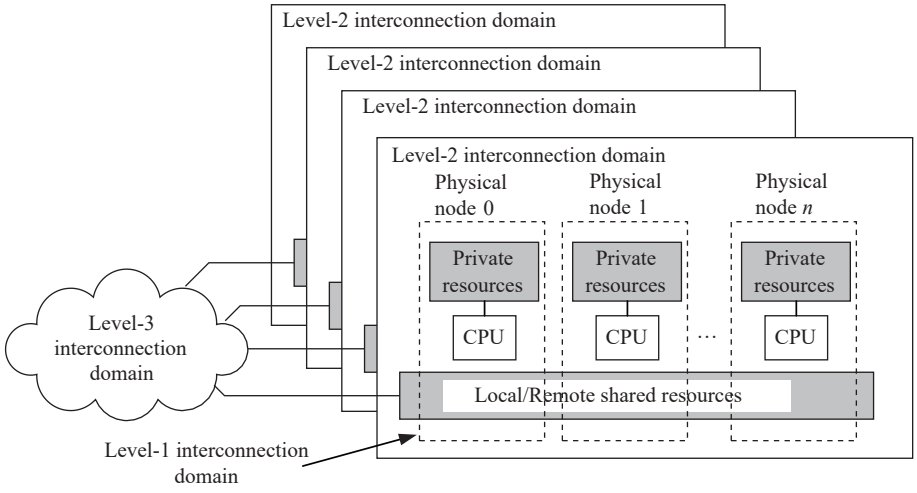

图 4 分布式云服务器结构

Fig.4 Distributed cloud server architecture

企业资源计划(Enterprise resource planning, ERP)系统．

三级互联：以二级互联域作为单元进行三级互联，在二级互联的基础上解决共享资源系统可扩展性问题．网络拓扑可采用 Mesh、Torus 等方式，网络协议可以采用公开协议，或者自定义的私有协议．通过该网络，一方面一个物理节点上处理器可以访问、共享三级互联域上任何一个节点的共享资源，实现全网区域的资源池化；另一方面根据系统的负载情况可以弹性的、无缝地减少或者添加三级互联的节点，以最小的代价缩小或者扩大云服务器的规模．此互联域非常适用于需要大量虚拟机平台、对计算性能要求不高、追求高性价比的业务场景．

1.2 云服务器体系结构的设计方法

根据如上所述三级互联资源的池化方法设计了一种分级硬件资源共享云服务器系统，如图 5 所示．

因为 CPU 和内存的配合对服务器性能的影响至关重要，现在的计算机体系结构一般把内存作为 CPU 的私有资源，与 CPU 一对一配置，所以把 CPU 和对应的内存组成一个物理节点放在一级互联域里．物理节点可以是单路 CPU，也可以是通过 SMP、NUMA 等形式组合成的多路 CPU．CPU 可为 x86 处理器、ARM 处理器、专用处理器以及其他架构处理器，可为高性能通用处理器(如 Intel 的 Xeon, AMD 的 Opteron)，也可为轻量级处理器(如 Intel Atom)，还可为一些专用处理器(如 AMD APU)．轻量级处理器的应用可使云服务器获得高的性能成本比、高的性能功耗比和高密度．

多个物理节点组成一个计算模组，对应一个二级互联域．物理节点具有异构性，不同的物理节点之间可以配置不同类型的 CPU，既可以使用追求高性能的通用处理器又可以使用追求高性价比的轻量级处理器，可根据云服务器使用的业务领域灵活配置．在具体实现上，可以把每个物理节点的存储总线(如 SATA(Serial advanced technology att-

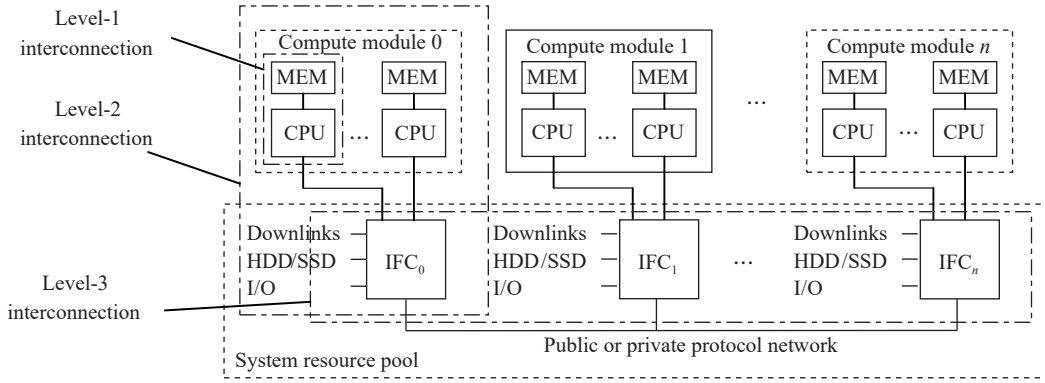

图 5　一种分级硬件资源共享云服务器系统

Fig.5　Cloud server based on a classified pooling hardware resource structure

achment) 总线)、网络总线(如千兆、万兆总线)、I/O 总线(如 PCIe 总线、USB(Universal serial bus) 总线、IIC(Inter-integrated circuit) 总线)连接到互联架构控制器(Interconnection fabric controller, IFC)上, 共享 IFC 上连接的存储资源、网络资源、I/O 资源, 从而实现硬件资源的池化.

作为云服务器系统核心的 IFC 设有以下三类互联通道:

(1)上行. 连接计算模组, 聚合不同物理节点的资源总线, 汇总连接到 IFC 相应的 Crossbar 总线上, 如存储、网络、I/O 等总线.

(2)下行. 连接网络、存储和 I/O 等物理设备, 为本地的计算模组和远端的计算模组提供共享资源.

(3)横向. 连接其他 IFC, 实现互联规模扩展, 互联协议可以采用公开的高速协议, 例如万兆以太网协议、IB(InfiniBand)协议, 或者自定义协议. 网络拓扑可以采用 2D Mesh、3D Torus 等架构, 任意物理节点上的 CPU 可以访问此网络上的共享资源.

2　云服务器原型系统的设计

根据上述提出的分布式硬件资源共享云服务器体系架构, 构建了如图 6 所示的云服务器原型系统架构, 用于云服务器概念与关键技术的验证. 此系统共有 16 个物理节点, 每 4 个物理节点组成一个计算模组, 计算模组之间采用 2D Mesh 网络拓扑互联. 图中 M0 表示标号为 0 的内存条; M1 表示标号为 1 的内存条; N0 ~ N3 表示标号为 0 ~ 3 的节点; Eth 表示以太网络.

原型系统的架构分为三级互联, 第一级互联域是一个物理节点, 采用一颗海光信息技术股份

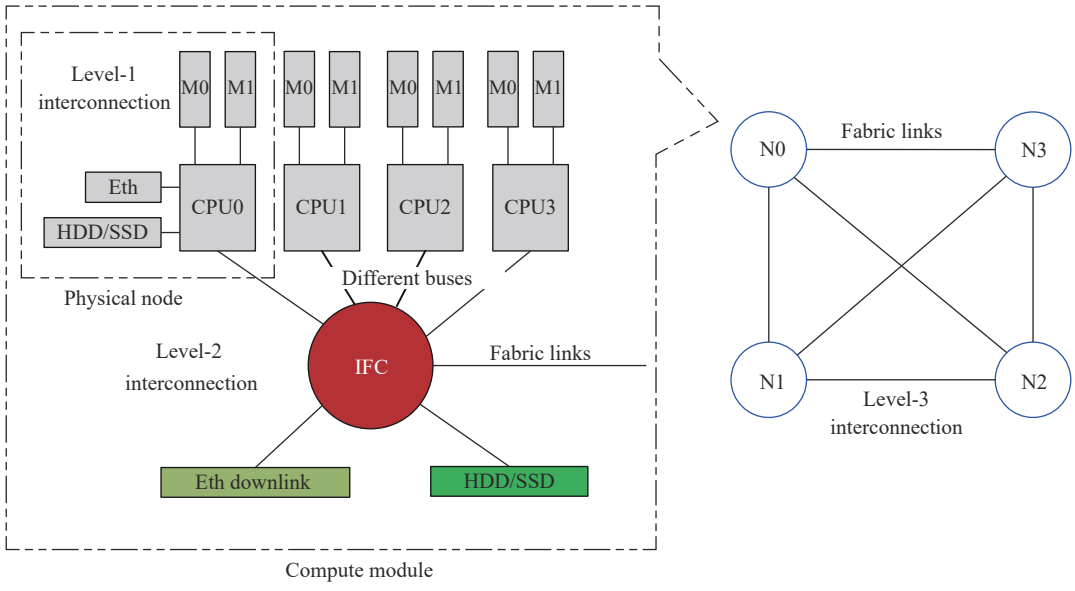

图 6　16 颗处理器云服务器原型系统架构

Fig.6　Architecture of the 16-processor cloud server prototyping system

有限公司型号为 3235 CPU，配置两条 DDR4 内存条.

第二级互联域是一个计算模组，包含四个物理节点，每个物理节点通过万兆、SATA、USB 总线采用星型拓扑结构聚合连接在 IFC Crossbar 总线上，共享 IFC 所连接的一个 SATA SSD（Solid state disk）硬盘和 100 G 以太网络.

第三级互联为四个 IFC 采用 2 × 2 Full Mesh 拓扑进一步互联，构建起 4 个计算模组共 16 颗 CPU 的云服务器原型系统. IFC 之间采用 100 G 以太网络协议进行互联.

IFC 采用 FPGA 芯片，FPGA[24–26] 具有灵活编程的特点，能模拟出各种物理接口和各种总线协议，为云服务器原型系统调试和验证提供便利.

按照图 6 所示的原型系统架构设计研发的云服务器原型系统样机如图 7 所示，样机主要由如下部分构成：

（1）四个计算模组：每个计算模组由四个物理节点组成，每个物理节点由一颗海光 3235 CPU 组成和 2 条 DDR4 内存组成；海光 3235 CPU 规格是 4 核、8 线程，主频 3.0 GHz，热设计功耗 (Thermal design power, TDP) 为 75 W；内存型号为 Samsung M393A4K40BB2，容量是 32 GB. 整个云服务原型系统共有 16 个物理节点，16 颗 CPU、32 条内存条、共 1 TB 内存容量. 计算模组运行的 OS 采用统信软件公司 UOS 桌面版，即 UOS-desktop-20-1031.

（2）物理节点：有一颗海光 3235 CPU，通过内存总线连接 2 条 DDR4 内存条；从 CPU 直接引出一个万兆以太网接口、一个 SATA3.0 总线、一个 USB2.0 总线连接 IFC 模块. 为了与 IFC 上共享资源的性能进行对比分析，在第一个物理节点，即 CPU0 直接连接私有的网卡和 SSD 硬盘进行性能测试，所测得值作为对比分析的基准，其他物理节点不用引出.

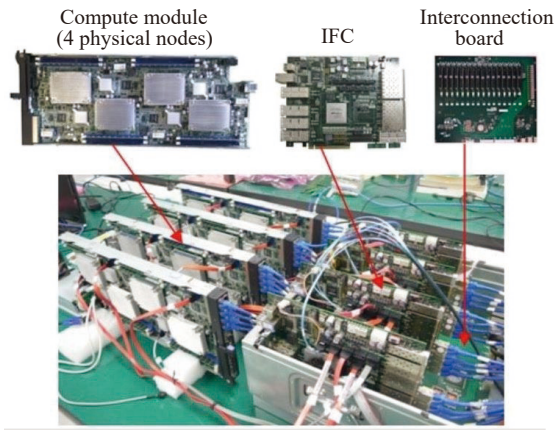

图 7 16 颗处理器的云服务器原型系统样品

Fig.7 Sample of the 16-processor cloud server prototyping system

（3）四个 IFC 模块：每个 IFC 模块基于一颗 FPGA 设计实现. IFC 有三个互联通道，上行连接物理节点的万兆以太接口、SATA 和 USB 总线，下行连接为计算模组提供共享的 100 G 以太网络接口和 SATA SSD 硬盘；横向通过 100 G 以太网络和其他 IFC 互联. FPGA 采用 Intel Stratix 10 GX 2800 系列（简称 GX 2800），具体型号为 1SG280HU2F 50E2LG，具有 93.3 万个可编程逻辑单元、32 对可达 17.4 Gbps 的全双工高速串行信号线和 64 对可达 28.3 Gbps 的全双工高速串行信号线，完全可以满足本设计的资源要求. 例如对于一个 100 G 以太接口大约需要 4.2 万个可编程逻辑单元模拟实现网卡控制器，需要 6 对 28.3 Gbps 全双工高速串行信号线，其中 4 对运行在 25 Gbps 来传输网络数据，另外 2 对作锁相环（Phase locked loop, PLL）使用. SATA 硬盘为 Samsung 公司的 6.0 Gbps 1.92 T SSD 硬盘，型号为 PM883-MZ7LH1T9HMLT.

（4）一个互联背板：支撑 IFC 模块 2 × 2 Full Mesh 互联.

云服务器原型系统实现了分布式硬件资源虚拟化共享模式，物理节点上的每个 CPU 可按需配置 IFC 上的存储资源和网络资源. 每个物理节点的 SATA 总线对应 IFC 内一个虚拟硬盘代理模块，硬盘代理模块利用 FPGA 的可编程逻辑单元模拟实现. 虚拟硬盘代理模块通过 XBAR 总线共享硬盘控制器. IFC 上挂接的 SATA SSD 硬盘划分为一个共享分区和 16 个私有分区，如图 8 所示，在共享分区中部署操作系统、应用软件，可被 16 个 CPU 共享，16 个 CPU 在启动或者运行中可从共享分区中读取操作系统或应用软件的镜像，各自加载在本地内存中独自运行，并把运行产生的数据写入到共享分区内，从而达到冗余资源减少、软件维护简便高效的目的. 私有分区为不同 CPU 私有，存取经不同的硬件虚拟化通道一一对应，数据安全性高. 共享分区和私有分区可按需分配、按需简约地灵活设计分区大小、所存数据的种类. 虚拟硬盘代理模块通过 IIC 总线与系统的管理芯片进行连接，一方面可以接收管理芯片的配置命令，对分区、数据包格式等参数进行设置；另一方面接受管理芯片的监控，反馈其运行的健康状态. 对于网络资源的共享模式，在 IFC 内部利用 FPGA 的可编程逻辑单元模拟实现每个物理节点的网卡控制器和一个网络交换机来实现每个物理节点和下行设

备的网络互联. IFC 上挂接的一个 100 G 以太网络接口作为下行接口, 可被所有的 CPU 节点共享使用, 可根据实际的网络需求灵活地调节对应物理节点的带宽.

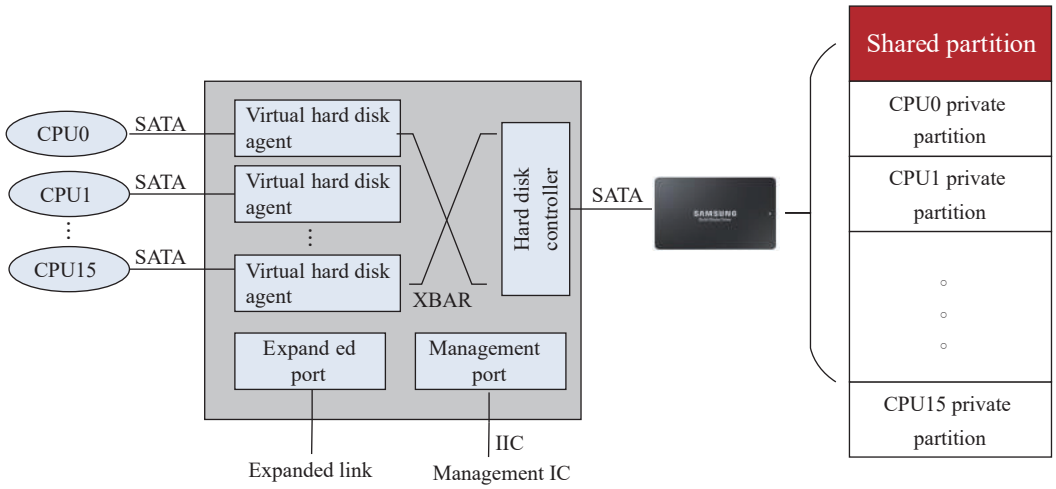

图 8　SATA 硬盘共享架构

Fig.8　Architecture of a SATA disk in shared mode

云服务器原型系统灵活地、低成本地实现了在存储、网络 I/O 等方面的硬件资源虚拟化, 达到了按需分配、按需简约的目的.

3　云服务器原型系统的性能评估

对云服务器原型系统的网络性能、存储性能进行测试分析, 评估分析硬件虚拟化性能和共享资源在三级互联系统中的性能损耗.

3.1　网络性能分析

从四个方面对云服务器原型系统的网络性能进行测试分析: 裸物理机网络性能, 硬件虚拟化网络性能, 纯软件虚拟化网络性能, 网络的聚合性能. 测试软件采用国际上公认的 benchmark 程序 Netperf V2.7.0 软件.

裸物理机网络性能的测试方法是利用物理节点 0 直连的一块 Intel 82599ES 万兆网卡连接一个外部压力机来直接测试物理节点 0 的万兆网络性能. 通过 Netperf 测试得到的三次实测数据的平均值为 1172 MBytes·s^{-1}, 并以此值作为对比分析的基准.

硬件虚拟化网络性能的测试方法是物理节点 0 的万兆网络接入 IFC, 经过网络的硬件虚拟化后, 利用 IFC 下行的 100 G 网口连接一个外部的压力机进行测试. 通过 Netperf 测试得到的三次实测数据平均值为 1168 MBytes·s^{-1}, 对比裸物理机测试结果发现硬件虚拟化的性能损耗是 0.34%. 这主要是因为 IFC 在硬件虚拟化过程中数据转换延迟、端口重映射带来的损耗, 可以忽略不计.

纯软件虚拟化网络性能测试是采用和裸物理机性能测试相同的硬件环境, 利用统信软件操作系统 UOS-desktop-20-1031 中的 KVM 功能创建 4 个物理核、64 GB 内存的虚拟机进行测试. 通过 Netperf 三次实测数据得到的平均值为 1098 MBytes·s^{-1}, 是裸物理机性能的 93.7%, 是硬件虚拟化性能的 94.0%. 因为 hypervisor 需要占用 CPU 资源运行, CPU 必须要在虚拟 CPU 环境与物理 CPU 环境下频繁地来回切换, 所以纯软件虚拟化网络性能有较大的损耗, 大约是 6%.

对于网络的聚合性能测试, 采用原型系统中 1 ~ 16 个物理节点通过 IFC 同时与下行 100 G 以太网口所连接的压力机传输数据进行测试. 为了分析在不同物理节点数量情况下 100 G 带宽的利用率, 需要选择一个基准, 利用 CPU0 直连一块 Mellanox 公司型号为 MCX556A 100G 网卡, 通过 Netperf 三次实测数据得到的平均值 11.7 GBytes·s^{-1} GBytes/s(即 11981 MBytes·s^{-1})作为对比分析的标准. 1 ~ 16 个节点下行链路的带宽使用情况如图 9 所示, 从 1 个节点到 10 个节点, 下行链路带宽随着物理节点数量的增加几乎线性增长, 一直到 11485 MBytes·s^{-1}, 此时带宽利用率为 95.9%, 然后随着物理节点数量的增加带宽出现轻微下降, 在 16 个物理节点时, 带宽为 11279 MBytes·s^{-1}, 利用率为 94.1%. 这说明在 1 ~ 10 个物理节点之间下行的 100 G 网络带宽足够宽裕, 一直到 10 个物理节点时才被用满. 超过 10 个物理节点, 由于下行带宽不足, 不同节点之间存在竞争造成了内部损耗, 所以带宽出现了下降, 但下降比较缓慢.

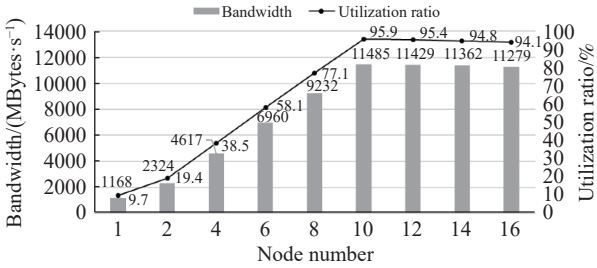

图 9 共享网络下行带宽

Fig.9 Bandwidth of the downlink when host accessing

因为 Netperf 被用来测试网络带宽的极限性能值，在平常使用过程中物理节点极少出现满负荷运行的情况，所以从以上分析可以看出，100 G 下行网络可有效支撑多个物理节点的网络共享，从 10 个到 16 个物理节点的带宽性能损失为 1.8%，损耗比较小．可利用本原型系统的弹性扩展能力，在同一个 IFC 上，即在第二级互联域上继续增加多个物理节点也不会引起太多的损耗，可进一步降低系统的成本．这表明本文设计的云服务器原型系统的网络池化设计工作正常，可以有效地降低了系统的复杂度．对比常规集群的搭建模式，此原型系统直接在 IFC 内实现网卡控制器和交换机功能，可不用再为每个物理节点设计单独的网卡控制器、互联的交换机和复杂的网线连接拓扑，系统简洁，节省了成本，提高了计算密度．此网络池化的架构，一方面根据每个节点带宽需求及优先级可配置不同的带宽，达到自动的负载均衡，充分地最大程度地利用已有网络资源避免浪费；另一方面，此架构具有弹性、可无缝拓展系统规模的优势，不但可以升级 IFC 下行网络，可采用更高速率的网络，而且也可以改变 IFC 横向互联的网络拓扑，可灵活采用更高维数、更多种类的网络拓扑连接更多的物理节点，实现真正的按需分配、按需简约的目的，非常适合云计算应用的特点．

3.2 存储性能分析

云服务器原型系统中 16 个物理节点共享 IFC 连接的一个 2.5 英寸 SATA 6.0 Gbps 1.92T SSD 固态硬盘，型号为 Samsung 公司的 PM883-MZ7LH1T 9HMLT，采用国际上公认的 benchmark 程序 FIO V3.12.2 测试存储性能．

从四个方面对云服务器原型系统的存储性能进行测试分析：裸物理机存储性能，硬件虚拟化存储性能，纯软件虚拟化存储性能，存储的聚合性能．

为了测试裸物理机存储性能，选择物理节点 0 直连的 SSD 硬盘进行测试，每项性能测试三次取平均值，得出 4 K 随机读 IOPS (Input/Output operations per second) 性能为 80832、4 K 随机写 IOPS 性能为 31873，并以此值作为对比分析的基准．

硬件虚拟化存储性能的测试方法是物理节点 0 的 SATA 总线接入 IFC，经 IFC 的硬件虚拟化后，利用 IFC 下行的 SATA 总线挂接 SSD 硬盘进行测试．通过 FIO 测试三次计算平均值，得出 4 K 随机读 IOPS 性能为 80387、4 K 随机写 IOPS 性能为 31616，分别是裸物理机性能的 99.4% 和 99.2%，对比裸物理机测试结果发现硬件虚拟化存储的性能损耗是大约 0.7%．这主要是因为 IFC 在硬件虚拟化过程中数据转换带来的损耗，对比真正的裸物理机性能，可忽略不计．

纯软件虚拟化存储性能测试是采用和裸物理机性能测试相同的硬件环境，在统信软件操作系统上创建 4 个物理核、64 GB 内存的虚拟机进行测试．通过 FIO 测试三次计算平均值，得出 4 K 随机读 IOPS 性能为 73476、4 K 随机写 IOPS 性能为 28206，分别是裸物理机存储性能的 90.9% 和 88.5%，是硬件虚拟化存储性能的 91.4% 和 89.2%．性能损失的原因同样是因为 hypervisor 占用了一部分 CPU 资源运行所致．

在存储的聚合性能测试中，原型系统中 1 ~ 16 个物理节点通过 IFC 同时与其下行接口所连接的 SATA SSD 硬盘进行 FIO 压力测试．选取上面的裸物理机的存储性能作为对比参考的标准．

图 10 为 16 个物理节点共享 IFC 上 SSD 硬盘 4 K 随机读、随机写的 IOPS 聚合性能，每项性能都测试三次取其平均值．

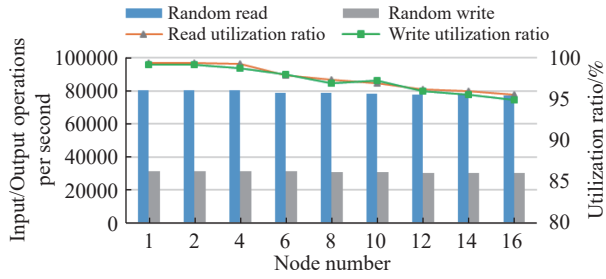

图 10 共享存储系统 IOPS 性能分析

Fig.10 Shared storage IOPS performance analysis

从图 10 中可以看出，对于随机读写性能来说，在 1 个节点时性能最高，4 K 随机读 IOPS 性能为 80387、4 K 随机写 IOPS 性能为 31616，随着节点数量的增加随机读写的性能缓慢地下降．在 16 个节点时随机读 IOPS 性能为 77247、利用率为 95.6%，随机写 IOPS 性能为 30253、利用率为 94.9%，也即 16 个节点 IOPS 聚合性能比裸物理机随机读写性

能分别损失了 4.4% 和 5.1% 的性能, 性能损失的原因是随着节点数量的增长, 共享资源的调度管理损耗了一部分性能.

从以上随机读和随机写的性能分析看, 在 1 ~ 16 节点下云服务器原型系统都能最大限度地发挥 IFC 上 SSD 硬盘性能, 存储聚合性能损失在 5% 左右. 总的来说, 存储的池化性能比较高、损耗可以忽略不计. 普通服务器配置的一块 SATA HDD(Hard disk drive)硬盘(如 Western Digital 公司型号为 HUS722T2TALA604 硬盘)的 4 K 随机读 IOPS 性能为 325、4 K 随机写 IOPS 性能为 303, 由此可以得出云服务器原型机的 SATA SDD 硬盘共享存储性能可完全满足近百个带有一块 SATA HDD 硬盘普通服务器节点的使用需求. 另外因为存储性能的测试是用 FIO 程序来测试极限性能值, 在平时使用过程中很少出现性能用满的情况, 所以 IFC 共享存储资源完全可以满足 16 个物理节点的使用需求, 并且可以根据物理节点的业务需求动态地分配存储容量和带宽, 达到按需简约、按需分配的云服务器设计要求.

从云服务器原型系统的网络性能和存储性能分析来看, 本文采用分布式硬件资源共享体系结构设计的云服务器原型系统一方面遵循按需简约、按需分配的设计理念, 真正实现了服务器硬件资源的池化、虚拟化, 可简易快捷地构建每个物理节点; 另一方面, 共享网络、存储的聚合性能损耗在 5% 左右, 该损耗对大部分应用来说是可接受的, 而且该原型系统的 IFC 还是基于 FPGA 器件来实现的, 未来如采用 ASIC 先进工艺实现, 共享损耗还会进一步的降低.

4　结论

(1)不同于软件虚拟化技术, 提出一种基于硬件虚拟化的云服务器系统架构, 此架构优点是能够实现系统硬件资源的共享, 硬件虚拟化速度比软件虚拟化速度更快, 占用的系统花销更少, 可以更加简洁地实现系统资源的虚拟化、池化, 做到按需分配、按需简约的目的, 与传统服务器相比最大限度提升服务器资源的使用效能.

(2)不同于常规的只能在单节点内进行资源的共享, 本文设计的云服务器架构将服务器的硬件资源按照三级互联的方式分别设计了计算池、内存池、存储池、网络池, 每级实现不同层次硬件资源的虚拟化, 可实现单节点、跨节点进而实现全区域的资源池化.

(3)本文设计的云服务器采用三级分层互联域, 对计算性能要求高的业务可在一级和二级互联域内运行, 对于追求虚拟机数量、性价比高的业务可在三级互联域内运行. 三级分层架构有效地扩展云服务器的适用范围.

(4)设计的云服务器架构具有异构性、可伸缩性. 不同种类的 CPU、存储、网络都可以无缝地融入到此云服务器内; 根据云计算需求规模的大小, 借助于此架构可以无缝地实现规模的变大或变小, 具有较好的弹性, 而不影响原先的架构和业务.

(5)研制了云服务器原型机, 为产品化提供了参考; 并实际测试了原型机的网络池化、存储池化的性能, 硬件虚拟化的损耗可忽略不计, 网络、存储的聚合性能损耗在 5% 左右, 证明本文设计的云服务器架构可显著地提高云计算的性价比.

参　考　文　献

[1] Gupta R. Above the clouds: a view of cloud computing. *Asian J Res Social Sci Humanities*, 2012, 2(6): 84

[2] Russinovich M, Costa M, Fournet C, et al. Toward confidential cloud computing. *Commun ACM*, 2021, 64(6): 54

[3] Fellah H, Mezioud C, Batouche M C. Mobile cloud computing: Architecture, advantages and security issues // *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*. Marrakech, 2020: 1

[4] Duan W X, Hu M, Zhou Q, et al. Reliability in cloud computing system: A review. *J Comput Res Dev*, 2020, 57(1): 102
(段文雪, 胡铭, 周琼, 等. 云计算系统可靠性研究综述. 计算机研究与发展, 2020, 57(1): 102)

[5] Hu X D, Ke X M, Yin F, et al. Shenwei-26010: A high-performance many-core processor. *J Comput Res Dev*, 2021, 58(6): 1155
(胡向东, 柯希明, 尹飞, 等. 高性能众核处理器申威26010. 计算机研究与发展, 2021, 58(6): 1155)

[6] Muda J, Tumsa S, Tuni A M, et al. Cloud-enabled E-governance framework for citizen centric services. *J Comput Commun*, 2020, 8(7): 63

[7] Sargunam S S. Cloud computing-system implementation for business applications. *Circuits Syst*, 2016, 7(6): 891

[8] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Gener Comput Syst*, 2012, 28(5): 755

[9] Rupra S S, Omamo A. A cloud computing security assessment framework for small and medium enterprises. *J Inf Secur*, 2020, 11(4): 201

[10] Hua Z, Wang X. Cloud computing and the essentials of security management. *OALib*, 2016, 3(5): 1

[11] Chen X M, Jha N K. A 3-D CPU-FPGA-DRAM hybrid architecture for low-power computation. *IEEE Trans Very Large*

Scale Integr (*VLSI*) *Syst*, 2016, 24(5): 1649

[12] Rusek M, Dwornicki G. Swarm-like distributed algorithm for scheduling a microservice-based application to the cloud servers. *Electronics*, 2021, 10(13): 1553

[13] Hu F, Che S J. Establishment of the docker-based laboratory environment. *OALib*, 2019, 6(6): 1

[14] Wood T, Ramakrishnan K K, Hwang J, et al. Toward a software-based network: Integrating software defined networking and network function virtualization. *IEEE Netw*, 2015, 29(3): 36

[15] Mijumbi R, Serrat J, Gorricho J L, et al. Network function virtualization: State-of-the-art and research challenges. *IEEE Commun Surv Tutor*, 2016, 18(1): 236

[16] Alam I, Sharif K, Li F, et al. A survey of network virtualization techniques for internet of things using SDN and NFV. *ACM Comput Surv*, 2021, 53(2): 35

[17] Minhas U I, Russell M, Kaloutsakis S, et al. NanoStreams: A microserver architecture for real-time analytics on fast data streams. *IEEE Trans Multi Scale Comput Syst*, 2018, 4(3): 396

[18] Dutta H, Kissler D, Hannig F, et al. A holistic approach for tightly coupled reconfigurable parallel processors. *Microprocess Microsyst*, 2009, 33(1): 53

[19] Hu W W, Yang L, Fan B X, et al. An 8-core MIPS-compatible processor in 32/28 nm bulk CMOS. *IEEE J Solid State Circuits*, 2014, 49(1): 41

[20] Cheng K, Doddamani S, Chiueh T C, et al. Directvisor: Virtualization for bare-metal cloud // *Proceedings of the* 16*th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. Lausanne, 2020: 45

[21] Venkateswaran S, Sarkar S. Time-sensitive provisioning of bare metal compute as a cloud service // 2019 *IEEE* 12*th International Conference on Cloud Computing*. Milan, 2019: 447

[22] Zhang X T, Zheng X, Wang Z, et al. High-density multi-tenant bare-metal cloud // *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. Lausanne, 2020: 483

[23] Zhou S J, Prasanna V K. Accelerating graph analytics on CPU-FPGA heterogeneous platform // 2017 29*th International Symposium on Computer Architecture and High Performance Computing* (*SBAC-PAD*). Campinas, 2017: 137

[24] Zheng C M, Yao X X, Zhou F, et al. Adaption and implementation of server chipsets for the Loongson CPU. *Chin J Eng*, 2022, 44(7): 1244
(郑臣明, 姚宣霞, 周芳, 等. 龙芯处理器服务器芯片组的适配与实现. 工程科学学报, 2022, 44(7): 1244)

[25] Zhang W L, Chen M Y, Fan J P. Emulation and forecast of HPL test performance. *J Comput Res Dev*, 2006, 43(3): 557
(张文力, 陈明宇, 樊建平. HPL测试性能仿真与预测. 计算机研究与发展, 2006, 43(3): 557)

[26] Wang S, Qi F B, Gu H F, et al. Linpack parallel performance model and its prediction. *Comput Eng*, 2012, 38(16): 81
(王申, 漆锋滨, 谷洪峰, 等. Linpack并行性能模型及其预测. 计算机工程, 2012, 38(16): 81)